類推に基づくエキスパート割当：kNN 検索を用いた Mixture-of-Experts

呂 博軒 [1,2] 村上 聡一朗 [2] 上垣外 英剛 [2,3] 張 培楠 [2]
[1] 東京科学大学 [2] 株式会社サイバーエージェント [3] 奈良先端科学技術大学院大学
lyu@lr.first.iir.isct.ac.jp
{murakami_soichiro, zhang_peinan}@cyberagent.co.jp
kamigaito.h@is.naist.jp

概要

Mixture-of-Experts（MoE）アーキテクチャは，パラメトリックルータにより各トークンを少数のエキスパートへ疎に割当て，計算コストを抑えたままモデル容量を拡張する．一方で，ルータのパラメータは学習後に凍結されることが多く，分布シフト下ではエキスパート割当が不安定になりやすい．本稿では、検索拡張型エキスパート割当手法である **kNN-MoE** を提案する．本手法では，参照集合上で正解トークンの尤度を最大化するように推定した「最適エキスパート割当」をメモリに保存し，推論時に kNN 検索を用いて割当を再利用する．近傍の類似度から混合係数を算出し，類似事例が乏しい場合にはルータへ自動的にフォールバックする．3 つの MoE モデルで評価した結果，kNN-MoE は Zero-shot を一貫して上回り，パラメータ更新なしに SFT に匹敵する性能を達成した．

1 はじめに

大規模言語モデル（LLM）のスケーリングは，Transformer[1] と Mixture-of-Experts（MoE）アーキテクチャ [2, 3, 4, 5] の組み合わせにより進展してきた．MoE は疎な活性化により，計算コストを大きく増やさずにモデル容量を増やすことができる [5]．ルーティングネットワーク（以下ルータ）は，各トークンに対して少数のエキスパートを動的に有効化する [5, 6]．しかし，MoE モデルの性能はこのルータの判断の質に強く依存する．

標準的なルータは，隠れ状態からエキスパート割当を予測する軽量な分類器として学習されるが，学習後は凍結され，推論時に割当を調整できない．学習分布から外れた入力（例えば，パープレキシティが高い事例）では学習した判断規則を外挿するしかなく，非最適なエキスパートを選びうる [7, 8, 9]．この硬直性は，特に専門領域や分布外タスクで MoE モデルの潜在能力を制約する．

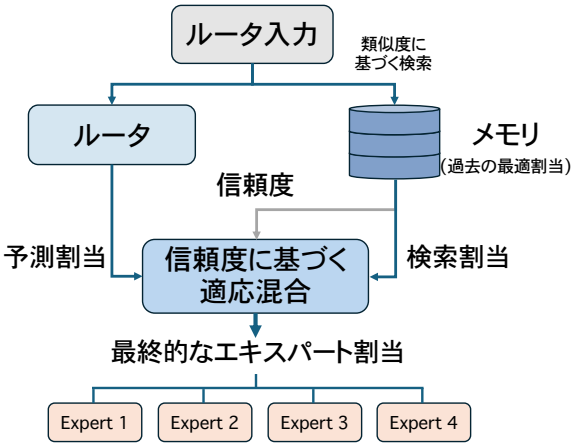

図 1 **kNN-MoE** の推論機構の概略図．入力に対し，ルータの予測とメモリから検索した近傍の割当を取得する．「信頼度に基づく適応混合」は近傍類似度に基づく係数で両者を補間し，最終的なエキスパート割当を出力する．

本稿では，ルータ入力の k 近傍 [10] を検索し，参照集合上で正解トークン尤度を最大化する割当を再利用してルータの判断を補正する **kNN-MoE** を提案する（図 1）．具体的には，ルータ入力（キー）と最適エキスパート割当（バリュー）を予めメモリに保存し，推論時に kNN 検索で近傍を取得する．さらに，近傍類似度に基づく信頼度係数で検索割当とルータの予測割当を補間し，類似度が低い場合はルータへフォールバックすることで検索ノイズを抑える．本アプローチは，パラメータ更新を伴わず，推論時にエキスパート割当を非パラメトリックに最適化する新たな枠組みを提示するものである．

3 種類の MoE モデル（OLMoE [11]，GPT-OSS [12]，Qwen3 [13]）に対し，GPQA [14]，SuperGPQA [15]，MMLU [16]，USMLE [17]，MedMCQA [18] で評価し

た．その結果，kNN-MoE は Zero-shot を上回り，計算コストの高い教師ありファインチューニング（SFT）に匹敵する性能を，モデルパラメータを更新せずに達成した．

2　関連研究

2.1　MoE におけるエキスパート割当の最適化

MoE におけるエキスパート割当の最適化がモデル性能を向上させることは，先行研究により示されている．例えば C3PO [9] は，モデルが正解した事例を用いて推論時にルーティングロジットを微調整する推論時適応（test-time adaptation）手法を提案したが，推論コストが大きい（ベースラインの約 5 倍）ことが課題となっている [9]．

同時期には，補助損失を用いてルータをファインチューニングし性能を改善する手法 [8] も提案されている．しかし，彼らの手法がルータのパラメータを恒久的に更新することに依存しているのに対し，本研究の提案手法はモデルパラメータを凍結したまま扱う非パラメトリックなアプローチである点で異なる．また，Su ら [7] は，入力テキスト上の自己教師あり損失を最小化することで，推論時にエキスパート割当を最適化する枠組みを提案した．これに対し，提案手法はメモリにキャッシュされた最適エキスパート割当を検索して利用するため，推論の度に最適化を行うオーバーヘッドを回避している．

2.2　非パラメトリック手法と検索拡張言語モデル

本研究は，外部データストアで生成モデルを補強する非パラメトリック手法と密接に関係する．代表例として kNN-LM [19, 20] は，キャッシュした隠れ状態から近傍トークンを検索し，最終出力分布を補間する．同様に，Retrieval-Augmented Generation（RAG） [21] はテキスト断片を検索して生成過程を条件付ける．さらに CBDT decoding [22] は，reranking で過去の高効用な選択を検索し，出力選択を導くことを検討した．これらが出力分布や生成内容を検索で補強するのに対し，本研究は MoE のエキスパート割当を検索で補正する点で異なる．

3　Mixture-of-Experts（MoE）

MoE Transformer [5] を L 層からなるモデルとして考える．このうち $\mathcal{L}_{\mathrm{MoE}} \subset \{1, \dots, L\}$ を MoE 層のインデックス集合とする．任意の $\ell \in \mathcal{L}_{\mathrm{MoE}}$ について，MoE モジュールは N 個のエキスパートネットワーク $\{E_i^{(\ell)}\}_{i=1}^{N}$ とルータ $R^{(\ell)}$ から構成される．

ℓ 層の MoE モジュールへの入力隠れ状態を $x^{(\ell)} \in \mathbb{R}^d$ とする．これは通常，直前の注意ブロックや正規化モジュールの出力である．MoE モジュールの要は，ルータが入力を関連するエキスパートに割当てる点にある．ルータは学習可能な重み行列 $W_r^{(\ell)} \in \mathbb{R}^{d \times N}$ でパラメータ化され，Top-K softmax により疎なゲーティング分布（エキスパート割当）$a^{(\ell)}(x^{(\ell)}) \in \mathbb{R}^N$ を予測する：

$$a^{(\ell)}(x^{(\ell)}) = \mathrm{TopK}\left(\mathrm{Softmax}(x^{(\ell)} W_r^{(\ell)})\right). \tag{1}$$

MoE モジュールの出力 $h^{(\ell)}$ は，予測された割当により選択されたエキスパートの出力を線形結合して得る：

$$h^{(\ell)} = \sum_{i=1}^{N} a^{(\ell)}(x^{(\ell)})_i \cdot E_i^{(\ell)}(x^{(\ell)}). \tag{2}$$

標準的な推論では，ルータ重み $W_r^{(\ell)}$ は凍結されたままである．この静的ルータは，テスト分布が学習分布から乖離した場合に適応性を制約しうる．

4　提案手法：kNN-MoE

本稿では，過去のルータ入力と最適エキスパート割当を予め「メモリ」に保持し，推論時にルータの予測割当を検索割当で補正する枠組み **kNN-MoE** を提案する（図 1）．kNN-MoE は，(1) **メモリ構築**（4.1 節）と，(2) **信頼度を考慮した適応混合**（4.2 節）の 2 段階からなる．本手法は各層で同一かつ独立に動作するため，以降は層の上付き添字 (ℓ) を省略する．

4.1　メモリ構築

メモリ構築では，層ごとにキー・バリューストア $\mathcal{M} = \{(k_t, v_t)\}$ を構築する．キー k_t にはルータ入力，バリュー v_t には正解トークン尤度を最大化する最適エキスパート割当を保存する．この過程は，(i) 参照集合からのデータ収集と，(ii) 最適エキスパート割当の導出からなる．

データ収集　参照集合 $\mathcal{D}_{\mathrm{ref}}$ の正解トークン系列 $\mathbf{y} = (y_1, \dots, y_T)$ を teacher forcing で与えて凍結 MoE モデルを実行し，各時刻 $t \in [1, T]$ でルータ入力 x_t と正解トークン y_t を収集する．ルータ入力を検索キーとして用い，$k_t = x_t$ とする．

最適エキスパート割当の導出　式 1 で得られる割当をそのまま保存するのではなく，正解トークン y_t の予測確率を最大化する最適エキスパート割当を求

める．ロジット $r \in \mathbb{R}^N$ から疎なエキスパート重みへの写像を $\pi(r) = \mathrm{TopK}(\mathrm{Softmax}(r))$ とする．トークン t に対して，パラメトリックロジット $x_t W_r$ の代わりに学習可能なロジットベクトル r_t を導入し，負の対数尤度を最小化する：

$$r_t^* = \arg\min_{r \in \mathbb{R}^N} \mathcal{L}_t(r), \tag{3}$$

$$\mathcal{L}_t(r) = -\log p_\theta\big(y_t \mid x_t, \text{routing}{=}\pi(r)\big), \tag{4}$$

ここで θ はネットワークの残り部分の凍結パラメータを表す．

式 3 は勾配降下法で解く．初期値として元のパラメトリックロジットを用い，$r_t^{(0)} = x_t W_r$ とする．その後，S ステップの更新を行う：

$$r_t^{(s+1)} = r_t^{(s)} - \eta \nabla_r \mathcal{L}_t(r_t^{(s)}), \quad \text{for } s = 0, \ldots, S-1, \tag{5}$$

ここで η は学習率である．この更新は全 MoE 層 $\ell \in \mathcal{L}_{\text{MoE}}$ のルーティングロジット $\{r_t^{(\ell)}\}$ に対して同時に適用され，各系列ごとに独立して最適化される．

最終的な最適エキスパート割当は $a^*(x_t) = \pi(r_t^{(S)})$ である．この値をメモリのバリューとして $v_t = a^*(x_t)$ と保存する．

以上より，メモリは次のように構成される：

$$\mathcal{M} = \{(x_t, a^*(x_t)) \mid \mathbf{y} \in \mathcal{D}_{\text{ref}},\ t \in [1, |\mathbf{y}|]\}. \tag{6}$$

この処理を $\mathcal{L}_{\text{MoE}}$ の各層について繰り返す．

4.2 信頼度を考慮した適応混合

推論時には，各 MoE 層でルータ入力 x に対し，4.1 節で構築したメモリから K 近傍 $\mathcal{N}(x) \subset \mathcal{M}$ を検索する．検索近傍を類似度で重み付けして集約し，メモリに基づく割当 $a_{\text{mem}}(x)$ を得る．検索された K 個のキー・バリューペアを $\{(k_j, v_j)\}_{j=1}^K$ とすると，

$$a_{\text{mem}}(x) = \sum_{j=1}^{K} \frac{s(x, k_j)}{\sum_{m=1}^{K} s(x, k_m)}\, v_j, \tag{7}$$

ここで $s(\cdot,\cdot)$ は類似度関数である．

次に，パラメトリックルータの予測割当 $a(x)$（式 1）と $a_{\text{mem}}(x)$ を融合するため，検索信頼度に基づく混合係数 $\lambda(x)$ を導入する．本稿では，近傍の平均類似度を信頼度として用いる：

$$\lambda(x) = \frac{1}{K} \sum_{j=1}^{K} s(x, k_j). \tag{8}$$

最終的なエキスパート割当は，両者を線形補間して得る：

$$a_{\text{final}}(x) = (1 - \lambda(x))a(x) + \lambda(x) a_{\text{mem}}(x). \tag{9}$$

最後に，$a_{\text{final}}(x)$ を用いて MoE 層の出力 h を計算する：

$$h = \sum_{i=1}^{N} a_{\text{final}}(x)_i \cdot E_i(x). \tag{10}$$

これにより，類似度が高い場合は検索割当を重視してルータの予測割当を補正し，類似度が低い場合は予測割当へフォールバックする．

5 実験

5.1 実験設定

データセット 一般推論ベンチマーク（GPQA, SuperGPQA, MMLU）と医療ベンチマーク（USMLE, MedMCQA）で kNN-MoE を評価した．各ベンチマークで，互いに重ならないテスト集合 $\mathcal{D}_{\text{test}}$ と参照集合 $\mathcal{D}_{\text{ref}}$ を用意し，$\mathcal{D}_{\text{ref}}$ は全手法で共有した．kNN-MoE はメモリ構築に，5-shot [23] は in-context 例の検索プールに，SFT は学習データに $\mathcal{D}_{\text{ref}}$ を用いた．各ベンチマークの分割とデータ規模は付録 A に示す．

モデル OLMoE（`OLMoE-1B-7B-0125-Instruct`）[11]，GPT-OSS（`gpt-oss-20b`）[12]，Qwen3（`Qwen3-30B-A3B-Instruct-2507`）[13] の 3 種類の MoE モデルを用いた．

ベースライン kNN-MoE を，計算コストの低い標準推論（**Zero-shot**，**5-shot**）と，SFT ベースの手法（**SFT**，**SFT（Router Only）**）と比較した．ここで，5-shot は参照集合 $\mathcal{D}_{\text{ref}}$ からテスト例の類似事例を 5 件検索して in-context 例に含める手法，SFT（Router Only）はルータパラメータのみをファインチューニングする手法である．なお，各ベースラインの詳細設定は付録 B に記載する．

評価方法 全データセットに対し，公式解答を用いて $\mathcal{D}_{\text{test}}$ 上の**正解率**（%）を報告する．

5.2 結果

3 種類のモデルにおける主要な結果を表 1 に示す．

Zero-shot および SFT（Router Only）との比較 全モデル・データセットで，kNN-MoE は Zero-shot と SFT（Router Only）を一貫して上回った．この結果は，パラメータ更新なしでも検索割当がルータのみより有効であることを示す．

5-shot との比較 kNN-MoE は 5-shot より安定して性能を改善した．Qwen3 では，5-shot と kNN-MoE の両方が Zero-shot を上回った一方で，GPT-OSS と OLMoE では 5-shot が Zero-shot を下回ることが多

表 1　3 つの MoE モデルにおける性能比較．太字は各モデル内で最良の性能を示す．

モデル	手法	GPQA	MMLU	SuperGPQA	USMLE	MedMCQA
	Zero-shot	27.27	46.77	13.02	32.81	35.57
	5-shot	21.72	33.06	11.09	31.31	25.36
OLMoE	SFT	21.72	46.89	**13.67**	**37.84**	34.78
	SFT（Router Only）	24.24	45.27	11.66	31.22	34.23
	kNN-MoE（提案）	**29.80**	**47.81**	13.27	35.04	**37.01**
	Zero-shot	43.94	70.20	23.52	67.29	56.20
	5-shot	36.87	63.72	19.95	60.02	52.76
GPT-OSS	SFT	41.92	**70.28**	**24.83**	65.61	56.06
	SFT（Router Only）	41.41	69.18	18.83	65.05	54.89
	kNN-MoE（提案）	**45.45**	**70.28**	24.35	**68.31**	**57.06**
	Zero-shot	41.41	78.59	34.85	75.30	62.51
	5-shot	44.44	**80.48**	35.56	77.17	66.58
Qwen3	SFT	39.90	79.08	**40.20**	**80.80**	66.60
	SFT（Router Only）	43.94	78.36	33.15	76.05	66.03
	kNN-MoE（提案）	**44.95**	78.86	35.15	76.70	**66.65**

かった．これは，in-context 例の連結により入力が大幅に長くなりモデルの性能が不安定になったこと，および in-context 例の検索ノイズが増えること（低類似事例の混入）が一因と考えられる．

SFT との比較　kNN-MoE は一部の設定で SFT を上回ったが，相対性能は条件に依存した．参照集合が小さい設定（GPQA と USMLE）では，SFT が Zero-shot を下回る場合があった．これは，少量データによる過学習の可能性がある．一方，kNN-MoE はデータ不足条件でも Zero-shot を安定して上回った．

5.3　考察: なぜ提案手法が有効か

本稿では，検索はパラメトリックルータが不確かなときほど有効であり，その不確かさはパープレキシティ（PPL）の増加と相関する，という仮説を立てた．これを検証するため，元モデル（Zero-shot）の PPL でテスト例を高／中／低の 3 群に分け，Zero-shot に対する kNN-MoE の正解率差を群ごとに算出した．表 2 に OLMoE での結果を示す．

表 2　OLMoE における正解率差（kNN-MoE − Zero-shot）を，元モデル（Zero-shot）の PPL で 3 群に分けて示す．

ベンチマーク	高 PPL	中 PPL	低 PPL
GPQA	+6.15	+2.99	-1.52
SuperGPQA	+0.17	+0.65	-0.06
MMLU	+2.07	+1.15	-0.13
USMLE	+6.32	+1.41	-1.13
MedMCQA	+6.04	+1.13	+0.56

高 PPL 例で改善が最大となり（例：GPQA で+6.15，USMLE で+6.32），全ベンチマークで同傾向が見られた．中 PPL 例でも改善は小さいが一貫して向上した．一方，低 PPL 例では改善がほぼゼロで，わずかに低下する場合もあった．これは，ルータ予測が十分なときに検索がまれにノイズを導入するためと考えられる．総合すると，高 PPL 入力ほどルータの割当予測が不確かになる一方で，kNN-MoE は類似事例の最適エキスパート割当でそれを補正できるため，PPL と改善幅が強く相関したと考える．

6　結論と今後の課題

本稿では，ラベル付き参照集合からオフラインで推定した最適エキスパート割当をメモリ化し，推論時の kNN 検索でルータの予測割当を補うエキスパート割当枠組み kNN-MoE を提案した．近傍類似度に基づく混合により，検索割当とルータの予測割当を補間してパラメータ更新なしにエキスパート割当を改善することが可能となった．3 つの MoE モデルと 5 つのベンチマークにおいて，kNN-MoE は Zero-shot および SFT（Router Only）を一貫して上回り，SFT に匹敵する性能を達成できることを示した．本手法の新規性は，静的ルータの予測割当を検索に基づく動的な割当で補間する点であり，MoE モデルの適応能力を拡張する新たな視点を提示する．

今後の課題として，メモリやキー（ルータ入力）の圧縮による検索高速化が挙げられる．例えば次元削減や量子化により距離計算を軽量化しつつ，精度の維持を目指す．第二に，ラベルなし参照集合への拡張である．**LLM-as-a-judge** [24] で参照集合に擬似ラベルを付与し，同様の尤度最大化で最適エキスパート割当を構築することを検討する．

参考文献

[1] Ashish Vaswani, Noam Shazeer, Niki Parmar, et al. Attention is all you need. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, **Advances in Neural Information Processing Systems**, Vol. 30. Curran Associates, Inc., 2017.

[2] Robert A. Jacobs, Michael I. Jordan, Steven J. Nowlan, and Geoffrey E. Hinton. Adaptive mixtures of local experts. **Neural Computation**, Vol. 3, No. 1, pp. 79–87, 03 1991.

[3] Michael I. Jordan and Robert A. Jacobs. Hierarchical mixtures of experts and the em algorithm. **Neural Computation**, Vol. 6, No. 2, pp. 181–214, 1994.

[4] Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. **arXiv preprint arXiv:1701.06538**, 2017.

[5] William Fedus, Barret Zoph, and Noam Shazeer. Switch transformers: scaling to trillion parameter models with simple and efficient sparsity. **J. Mach. Learn. Res.**, Vol. 23, No. 1, January 2022.

[6] Albert Q. Jiang, Alexandre Sablayrolles, et al. Mixtral of experts, 2024.

[7] Guinan Su, Yanwu Yang, Li Shen, Lu Yin, Shiwei Liu, and Jonas Geiping. Rewiring experts on the fly:continuous rerouting for better online adaptation in mixture-of-expert models, 2025.

[8] Zhongyang Li, Ziyue Li, and Tianyi Zhou. Routing manifold alignment improves generalization of mixture-of-experts llms, 2025.

[9] Zhongyang Li, Ziyue Li, and Tianyi Zhou. C3PO: Critical-layer, core-expert, collaborative pathway optimization for test-time expert re-mixing. In **Second Conference on Language Modeling**, 2025.

[10] B.V. Dasarathy. **Nearest Neighbor (NN) Norms: Nn Pattern Classification Techniques**. IEEE Computer Society Press tutorial. IEEE Computer Society Press, 1991.

[11] Niklas Muennighoff, Luca Soldaini, et al. Olmoe: Open mixture-of-experts language models, 2024.

[12] OpenAI. gpt-oss-120b & gpt-oss-20b model card, 2025.

[13] Qwen Team. Qwen3 technical report, 2025.

[14] David Rein, Betty Li Hou, Asa Cooper Stickland, et al. GPQA: A graduate-level google-proof q&a benchmark. In **First Conference on Language Modeling**, 2024.

[15] M-A-P Team. Supergpqa: Scaling llm evaluation across 285 graduate disciplines, 2025.

[16] Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. **Proceedings of the International Conference on Learning Representations (ICLR)**, 2021.

[17] Di Jin, Eileen Pan, Nassim Oufattole, et al. What disease does this patient have? a large-scale open domain question answering dataset from medical exams. **arXiv preprint arXiv:2009.13081**, 2020.

[18] Ankit Pal, Logesh Kumar Umapathi, and Malaikannan Sankarasubbu. Medmcqa: A large-scale multi-subject multi-choice dataset for medical domain question answering. In Gerardo Flores, George H Chen, Tom Pollard, Joyce C Ho, and Tristan Naumann, editors, **Proceedings of the Conference on Health, Inference, and Learning**, Vol. 174 of **Proceedings of Machine Learning Research**, pp. 248–260. PMLR, 07–08 Apr 2022.

[19] Urvashi Khandelwal, Omer Levy, Dan Jurafsky, Luke Zettlemoyer, and Mike Lewis. Generalization through memorization: Nearest neighbor language models. In **International Conference on Learning Representations**, 2020.

[20] Frank F. Xu, Uri Alon, and Graham Neubig. Why do nearest neighbor language models work? In **Proceedings of the 40th International Conference on Machine Learning**, ICML'23. JMLR.org, 2023.

[21] Patrick Lewis, Perez, et al. Retrieval-augmented generation for knowledge-intensive nlp tasks. In **Proceedings of the 34th International Conference on Neural Information Processing Systems**, NIPS '20, Red Hook, NY, USA, 2020. Curran Associates Inc.

[22] Hiroyuki Deguchi and Masaaki Nagata. Case-based decision-theoretic decoding with quality memories. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng, editors, **Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing**, pp. 33679–33694, Suzhou, China, November 2025. Association for Computational Linguistics.

[23] Tom B. Brown, Mann, et al. Language models are few-shot learners. In **Proceedings of the 34th International Conference on Neural Information Processing Systems**, NIPS '20, Red Hook, NY, USA, 2020. Curran Associates Inc.

[24] Lianmin Zheng, Wei-Lin Chiang, Sheng, et al. In **Proceedings of the 37th International Conference on Neural Information Processing Systems**, NIPS '23, Red Hook, NY, USA, 2023. Curran Associates Inc.

[25] Yanzhao Zhang, Mingxin Li, Dingkun Long, et al. Qwen3 embedding: Advancing text embedding and reranking through foundation models. **arXiv preprint arXiv:2506.05176**, 2025.

[26] Edward J Hu, Yelong Shen, Phillip Wallis, et al. LoRA: Low-rank adaptation of large language models. In **International Conference on Learning Representations**, 2022.

[27] Jeff Johnson, Matthijs Douze, and Hervé Jégou. Billion-scale similarity search with GPUs. **IEEE Transactions on Big Data**, Vol. 7, No. 3, pp. 535–547, 2019.

A　評価ベンチマークの統計

本研究で用いた各ベンチマークについて，テスト分割と参照分割の対応，およびサンプル数を表 3 にまとめる．ここで k は 10^3 を表し，本文 5.1 節で述べた通り $\mathcal{D}_{\text{test}}$ と $\mathcal{D}_{\text{ref}}$ は互いに重ならないように構成した．参照集合の規模はベンチマーク間で大きく異なり，特に GPQA と USMLE では $|\mathcal{D}_{\text{ref}}|$ が小さいため，手法間のサンプル効率の差が現れやすい．

表 3　評価ベンチマークの統計．$\mathcal{D}_{\text{test}}$ は精度の報告に用いる．$\mathcal{D}_{\text{ref}}$ は手法間で共有する参照分割であり，kNN-MoE のメモリ構築，5-shot における例の検索，および教師ありファインチューニング（SFT）の学習に用いる．

ベンチマーク	テスト分割	参照分割	$\lvert\mathcal{D}_{\text{test}}\rvert$	$\lvert\mathcal{D}_{\text{ref}}\rvert$
GPQA	Diamond	Main（filtered）	0.20k	0.20k
MMLU	Test subset	Train subset	14.00k	1.81k
SuperGPQA	Held-out split	Random subset	23.50k	3.00k
USMLE	Test subset	Train subset	1.27k	0.20k
MedMCQA	Test subset	Train subset	6.15k	1.00k

B　ベースラインの詳細設定

本稿で比較に用いたベースラインの設定は以下の通りである．

- **Zero-shot:** ルータ $R^{(\ell)}$ を凍結したまま，MoE モデルを Zero-shot QA プロンプトで推論した．
- **5-shot:** 各テスト例に対し，参照集合 $\mathcal{D}_{\text{ref}}$ から質問埋め込みのコサイン類似度が最も高い 5 例（質問と正解）を検索し，テスト質問の前に連結して in-context プロンプトとした．質問埋め込みには `Qwen3-Embedding-0.6B` [25] を用いた．
- **SFT:** 標準的なクロスエントロピー損失で $\mathcal{D}_{\text{ref}}$ を学習し，モデルの全線形層に LoRA アダプタ [26] を適用した．最大エポック数は 3 とし，$\mathcal{D}_{\text{ref}}$ を 85%の学習分割と 15%の検証分割に分け，検証損失が最小のチェックポイントを採用した．学習時・推論時ともに Zero-shot QA プロンプトを用いた．
- **SFT（Router Only）:** ルータパラメータ $\{W_r^{(\ell)}\}_{\ell\in\mathcal{L}_{\text{MoE}}}$ のみを $\mathcal{D}_{\text{ref}}$ 上でファインチューニングし，それ以外のパラメータは凍結した．学習プロトコルとプロンプトは SFT と同一とした．

C　kNN-MoE の実装詳細

検索には FAISS[27] を用い，MoE 層ごとに 1 つのインデックスを構築した．キーとして当該層のメモリ $\mathcal{M}$ に保存されたルータ入力 $\{x_t\}$ を用いた．近傍数 K は全層で共有し，$K = 1$ とした．学習率は $\eta = 2\times10^{-2}$，各トークンあたりの勾配降下ステップ数は $S = 1$ とした．類似度関数には RBF カーネル $s(x,k) = \exp(-\gamma\|x-k\|^2)$ を用い，γ はメモリ内の平均最近傍距離に基づき設定した．

D　C3PO との比較

表 4　OLMoE モデルにおける推論時エキスパート割当適応手法 C3PO との比較．kNN-MoE は小規模データ（GPQA，USMLE）で高いサンプル効率を示す．

手法	GPQA	MMLU	USMLE
Zero-shot	27.27	46.77	32.81
C3PO	24.24	**48.04**	32.93
kNN-MoE	**29.80**	47.81	**35.04**

kNN-MoE を，推論時エキスパート割当適応の代表手法 C3PO [9] と比較する．表 4 は，参照集合サイズが異なる 3 ベンチマークにおける OLMoE の正解率を示す．参照データが限られるベンチマーク（GPQA，USMLE，$|\mathcal{D}_{\text{ref}}| \approx 0.20$k）では，C3PO の改善が小さい（あるいは低下する）一方で，kNN-MoE は一貫して改善した．一方，参照集合が比較的大きい MMLU（$|\mathcal{D}_{\text{ref}}| \approx 1.81$k）では，C3PO が kNN-MoE を上回った．

この差はサンプル効率の違いに起因すると考えられる．C3PO はベースモデルが正答した参照例のみでルータを更新するため，参照集合が小さい場合には有効な学習信号が減りやすい．対照的に kNN-MoE は，正誤に関わらず正解トークン尤度を最大化する割当を用いるため，より多くの例から監督信号を得られる．